MetaRule

Path Management

Sean Barnum, Cigital, Inc. [vita[1]]

2007-03-29

Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 4593 bytes

Attack Category	• Malicious Input
Vulnerability Category	• Buffer Overflow
Software Context	• File Path Management
Location	
Description	Output buffers for path functions must be sized to hold at least MAX_PATH characters. In general, Windows path functions need to have a buffer of at least MAX_PATH otherwise they are subject to buffer overruns. For some Windows functions, if a Unicode path begins with "\\?" then the path can be longer than MAX_PATH characters, up to 32,000+ characters in length. It is unclear when this applies in a way that would lead to buffer overflows.
APIs	

Function Name	**Comments**
PathAddBackslashA	
PathAddBackslashW	
PathAddExtension	
PathAddExtensionA	
PathAddExtensionW	
PathAppend	
PathAppendA	
PathAppendW	
PathCanonicalize	
PathCanonicalizeA	
PathCanonicalizeW	
PathCombine	
PathCombineA	
PathCombineW	

1. http://buildsecurityin.us-cert.gov/bsi-rules/35-BSI.html (Barnum, Sean)

<table>
<tr><td>Method of Attack</td><td></td></tr>
<tr><td>Exception Criteria</td><td></td></tr>
<tr><td>Solutions</td><td>
<table>
<tr><th>Solution Applicability</th><th>Solution Description</th><th>Solution Efficacy</th></tr>
<tr><td>Path functions.</td><td>Ensure that output buffer is sized as MAX_PATH characters.</td><td>Effective</td></tr>
</table>
</td></tr>
<tr><td>Signature Details</td><td>Any function that returns a path as a result.</td></tr>
<tr><td>Examples of Incorrect Code</td><td><pre>WCHAR path[] = L"MyFile.dat"; // Buffer is too small
LPWSTR pszPath = path;
LPCWSTR dirs[] = { NULL };

if (!PathResolve(pszPath, dirs, PRF_VERIFYEXISTS)
{ handleError(); }</pre></td></tr>
<tr><td>Examples of Corrected Code</td><td><pre>WCHAR path[MAX_PATH] = L"MyFile.dat"; // Buffer is correctly sized
LPWSTR pszPath = path;
LPCWSTR dirs[] = { NULL };

if (!PathResolve(pszPath, dirs, PRF_VERIFYEXISTS)
{ handleError(); }</pre></td></tr>
<tr><td>Source Reference</td><td>• Rough Auditing Tool for Security (RATS)[2]</td></tr>
<tr><td>Recommended Resource</td><td></td></tr>
<tr><td>Discriminant Set</td><td>
<table>
<tr><td>Operating System</td><td>• Windows</td></tr>
<tr><td>Languages</td><td>• C
• C++</td></tr>
</table>
</td></tr>
</table>

Cigital, Inc. Copyright

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

1. mailto:copyright@cigital.com